Niha Imam

02/12/2020

CS 484 HW 1 – Amazon Review Classifier

Username: iateyourcookie

Rank: 10

Accuracy: 0.78

How to run the program

Libraries utilized: sklearn and numpy

The test and the train data file must be

My approach/train of thought

Initially I did not know what I was doing at all. My first approach was to convert both training and testing data into lists. Then take the index of the review I needed to predict then go to the training list and check all the neighbors which gave me an accuracy of 0.62 which wasn’t bad, but I don’t think data mining was involved in it at all. It involved multiple for loops and the run time was six hours.

My second approach was to take the two documents and read all the lines. After that takeout words that were too short and cleaned the data. After that I proceeded to take the unique words from both lists and combined them in a big list but the run time for traversing a big list was an hour, so I googled, and piazza and stack exchange recommended implementing a csr matrix. I used csr matrix from train and test data using a feature list and normalized it and calculated the cosine similarity and then calculated the nearest neighbors. If implemented properly the runtime was supposed to be reduced but my implementation of the matrix was not optimized.

The last approach was to read in the train file and the test file. I tried to use the bag of words model by utilizing count vectorizer to tokenize text document and build a vocabulary of known words. Then fit\_transfrom the train data in order to learn and encode and then transform to encode. Then the cosine similarity value was calculated using the transformed data. For every value in the cosine similarity value looked for the neighbors using the assigned k value of 77. For those indexes I kept track of the sentiment by starting at 0 and adding 1 to the sentiment counter if the review was positive and subtracting 1 to the sentiment counter if the review was negative. In the end if the sentiment counter was >0 the review was positive and if the sentiment counter was <0 the review was negative. Since the k value chosen was odd the sentiment value would never go back to 0